IN-CYCLE SHUFFLE

BACKGROUND OF THE INVENTION

[0001] A Boiling Water Reactor (BWR) typically has a core formed of several hundred fuel bundle assemblies. Each fuel bundle assembly is formed of several fuel assembly rods, and each fuel assembly rod contains a variety of radioactive elements. Typically a number of different fuel bundle assemblies are created, and a core designer creates a core design using the different fuel bundle assemblies. The core design involves establishing the positioning of each bundle in the core, which is called the loading map, and establishing an operation plan over a period of time referred to as the operation or loading cycle. The operation plan establishes the operating parameters of the core over the loading cycle. Operating parameters (also referred to interchangeably herein as "independent control variables" and "design inputs") include, for example, various physical component configurations and controllable operating conditions that may be individually adjusted or set (e.g., control rod positions and adjustments over time - typically referred to as the rod pattern). The loading cycle is typically designed for 1 year, 1.5 years or 2 years. Once a core design has been created, the core design must be licensed by NRC before being put into operation.

[0002] Once licensed, the core design can be put into practice. The start of a period of operation is typically referred to as BOC, Beginning of Cycle. When the reactor starts, flow and power are slowly increased over a period of a couple days. Eventually, the reactor is said to be at rated operating conditions (rated power, flow, inlet enthalpy, core pressure, etc.). Here, the reactor will stay at rated operating conditions. After a period of time ranging from a few days to several months the reactor must change the operating or control rod patterns in order to recalibrate for changing core reactivity or in order to operate in an alternative rod pattern strategy or sequence. In order to minimize duty on the fuel bundle assemblies, power is typically reduced during these rod pattern modifications. Hence, additional power ascensions are required after each rod pattern modification. This method of operation occurs for the loading cycle (1 year, 1.5 year, or 2 year periods) until the end of reactivity and operation occurs. The ending of a plants loading cycle is typically referred to as EOC, End of Cycle.

SUMMARY OF THE INVENTION

[0003] The inventors have discovered that planning a shutdown and reshuffle process into the initial operating plan of a reactor core may provide the beneficial result of increased energy output and thus increased revenue over operating plans with no shutdown and shuffling process. Such planned outage could also be utilized to mitigate any unanticipated issues that might occur during the cycle. Such issues may include: failed fuel, or equivalently, a 'leaking' fuel rod; channel bow, a deformation of the fuel bundle channel due to non-uniform bundle exposure; and excess control blade history - a well-known nuclear reaction induced phenomenon that can limit the thermal performance of the controlled bundles in the current and subsequent fuel cycles. Mitigation of such problems will proactively improve plant performance with regard to energy produced and plant availability.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] The present invention will become more fully understood from the detailed description given herein below and the accompanying drawings, wherein like elements are represented by like reference numerals, which are given by way of illustration only and thus are not limiting on the present invention and wherein:

[0005] FIGURE 1 A is a block diagram illustrating a system for the optimization of multiple operational control-variables for a nuclear reactor;

[0006] FIGURE 1B is a schematic illustration of an example network arrangement of independent processors in which the present invention may be embodied;

[0007] FIGURE 2 is a data flow diagram illustrating the basic data flow between processes in an example embodiment of a software system for implementing the reactor core multiple control-variable optimization;

[0008] FIGURE 3 is a block diagram illustrating an example embodiment of a software system for implementing the reactor core multiple control-variable optimization method;

[0009] FIGURE 4 is a flowchart illustrating exemplary functional program control steps performed by a response surface initialization module;

[0010] FIGURE 5A is a flowchart illustrating functional program control steps performed by a fuel bundle loading module;

[0011] FIGURE 5B is a flowchart illustrating exemplary functional program control steps performed by a control rod axial positioning module;

[0012] FIGURE 5C is a flowchart illustrating exemplary functional program control steps performed by a core flow module;

[0013] FIGURE 5D is a flowchart illustrating exemplary functional program control steps performed by a sequence interval module;

[0014] FIGURE 5E is a flowchart illustrating exemplary functional program control steps performed by an fuel bundle characteristics module;

[0015] FIGURE 6 is a flowchart illustrating exemplary functional program control steps performed by an polynomial coefficient development module;

[0016] FIGURE 7 is a flowchart illustrating exemplary functional program control steps performed by an polynomial usage module;

[0017] FIGURE 8 is a flowchart illustrating exemplary functional program control steps for saving and modifying response surface results;

[0018] FIGURE 9 illustrates a flow chart of the method of improving reactor performance through in-cycle shuffling according to an embodiment of the present invention; and

[0019] FIGURE 10 illustrates a flow chart of the method of improving reactor performance through in-cycle shuffling according to another embodiment of the present invention.

DETAILED DESCRIPTION OF THE EMBODIMENTS

[0020] First a method of developing an operating plan will be described in detail. Then, the method of improving reactor performance through in-cycle shuffling according to the present invention will be described.

Creating An Operating Plan

[0021] The following description is directed toward an exemplary embodiment for creating a response surface. The methodology for creating the response surface may be operative as an end-user application running, for example, under the Microsoff^{e} Windows 95/NT environment. However, creation of the response surface is not limited to any particular computer system or any particular environment. Instead, those skilled in the art will find that the system and methods presented herein may be advantageously applied to environments requiring management and/or optimization of any multiple control-variable critical industrial/scientific process or system, including chemical and mechanical process simulation systems, pressurized water reactor simulation systems, boiling water reactor simulation systems, and the like. Moreover, the system may be embodied on a variety of different platforms, including UNIX, LINUX, Macintosh, Next Step, Open VMS, and the like. Therefore, the description of the exemplary embodiments which follows is for purposes of illustration and not limitation.

[0022] Referring first to FIGURE 1A, a block diagram illustrates an example system embodiment for optimization of multiple operational control-variables or design inputs for a nuclear reactor. Reactor plant specific design constraints and cycle specific initial data, 1, defining a particular reactor core, 3, are provided as input data to the optimization system 2. Optimized values for operational control variables or design inputs (e.g., rod pattern, fuel loading, core flow, etc.) are provided as outputs for use in the design and management of the nuclear reactor core.

[0023] Referring to FIGURE 1B, an example computer network arrangement is shown on which the optimization method that includes creating a response surface may be embodied. A plurality of general purpose computers/processors 10/11 are coupled to a local area communications network (LAN) 15, which may itself be coupled to one or more distinct open or private access network(s) 20 for communications with one or more remote computers 21. In a preferred embodiment, the multiple control-variable optimization method is implemented via software modules resident on at least one of computers 10. As explained below, the modules may be distributed among computers 10 or may be resident on one or more of computers 10 (and 21) that communicate via LAN 15 and/or network(s) 20.

[0024] As represented in FIGURE 1B, communications network 15 and/or 20 can be an open network, such as the Internet, or a private access network, such as a local area network (LAN) or a wide area network (WAN). General purpose computers 10 are coupled directly or via a modem to network 15 and consist of independent processor 11 with or without dedicated memory 12 in addition to conventional I/O and user interface components (not shown). Computers 10 can be any of a variety of high speed processors, for example, a VMS-Alpha computer system, a Legacy computer system, a high-speed work station or a high-speed compatible personal computer (such as a desk-top or laptop system). Communications over the networks 15 and 20 can be accomplished using any preferred combination of conventional and proprietary protocols that facilitates efficient inter-processor communications such as, for example, the TCP/IP protocol.

[0025] Two or more of computers10 (21), preferably systems that are capable of supporting the execution of appropriate software for the simulation of nuclear reactor core operations, are coupled via some communications link(s) such as LAN 15 and/or network 20 for exchanging data files and control information. Most any conventional reactor core simulation program (or suite of programs), such as for example, General Electric's (GE's) "PANACEA" 3-D reactor core simulation program, may be used in conjunction with the present invention. This type of simulator program is capable of processing three dimensional variables defining the core. An input file containing values for selected "independent" reactor control-variables or design inputs (e.g., fuel loading, rod pattern, core flow, etc.) is provided as an input and the simulator program provides an output file comprising values for selected performance parameters or operational outputs. For example, the operational outputs include but are not limited to parameters conventionally used to gauge reactor core performance over the fuel operating cycle, such as critical power ratio (CPR), shutdown margin (SDM), maximum average planar linear heat generation rate (MAPLHGR), maximum fraction of linear power density (MFLPD), Hot excess reactivity, radial and axial power peaking, peak fuel rod and bundle exposure, Uranium utilization as measured by reactor energy output produced (in mega-watt-days) per kilogram of Uranium-235 loaded, etc.

[0026] Many of the performance parameters analyzed are both spatially and time dependent, such as, for example, MAPLHGR, MFLPD, and minimum critical power ratio (MCPR). Accordingly, some of these operational outputs may be indicative of the state of the reactor core at a plurality of discrete intervals (i.e., each and every "exposure step") throughout one or more core refueling cycles.

[0027] Referring now to FIGURE 2, the basic functional processes and data flow within an example software system 200 for implementing the multiple control-variable optimization method, which creates the response surface, are described. Information concerning a selectable "resolution" level (explained in greater detail below), other processing options and the reactor core cycle-specific input data information is preferably input by the user at an initial stage (not shown). A cycle-specific reactor core profile input file 201, containing reactor core characteristics and operational critical-to-quality constraints specific to a particular reactor plant for a particular fuel-cycle, is built from this user-input information. The cycle-specific input data is used to identify initial independent control-variable or design input values that define an initial "center-point" data case for a particular reactor. This center-point data is provided as an input data file 202 to a reactor core simulation program (actual simulation program not shown). A reactor core operation simulation 207 is conducted using the center-point data. For example, a three-dimensional (3-D) analysis core simulation is performed on a selected "host" computer 10. When the simulation process is complete, a center-point case simulation output data file 212 is produced. The center-point case simulation output data from this file is then stored in a multidimensional array within the digital storage memory of the selected "host" computer 10 and is used as the basis for creating a type of response surface 219 for evaluating the reactor performance for different control-variable values.

[0028] Next, separate simulations of the same reactor core operating under different physical conditions and constraints represented by predetermined changes in independent control-variable values for selected operational control variables are conducted contemporaneously by the software system. Different simulator input data files 203-206 are created, each reflecting a change in a value for a selected control-variable (i.e., design input), and each input file is submitted to an independent reactor core simulator program or process 208 - 211 resident on one or more independent computers or processors 10,21 connected via the communications network 15,20. After performing a core simulation based on the values in the received input file, each simulator process returns an output data file 213-216 reflecting the resultant output values of the dependent variables (i.e., operational outputs) of the reactor core. Once all of the reactor core simulations for each of the independent variable cases 208-211 are complete, the data from simulator output files 213-216 is normalized as indicated at block 217, for example, by dividing each data item by output data obtained from the original "center-point" case 212.

[0029] After all the simulation case output data is normalized, the normalized data for each independent control-variable case is characterized as a transfer function. For example, the normalized data is mapped to a set of corresponding second-order polynomials reflecting the change in a given simulator output with respect to a change in a given control variable; however, polynomials of higher or lesser orders may be used. In other words, second-order polynomials, each of which is characterized by a set of associated polynomial coefficients, are selected to fit the simulation output data obtained in a few limited number of reactor core simulations. For instance, three simulations are exemplary used for evaluating each independent control-variable: a center-point case and two variation cases; wherein the center-point case quantitative value for the particular control-variable is respectively incremented and decremented. The polynomials are then utilized as "predictors" to predict quantitative values of selected operational outputs (i.e., performance parameters) for each control-variable. Coefficients which uniquely define each polynomial are developed from the normalized simulator output data, as indicated at block 218, using conventional algorithmic techniques for solving second-order polynomials (e.g., curve fitting). This normalized coefficient data is stored in an area of computer memory defined herein as the "response surface", as represented by block 219. Basically, response surface 219 contains the dependent operational output (performance parameter) response or relationship of the reactor to individual or combined changes in values of the design input (control-variables). In this manner, the response surface serves as sort of a cyber-workspace and data-array repository for storing the resultant reactor core simulation output data from different case simulations for multiple independent control-variables.

[0030] Next, the polynomials for each control-variable are evaluated 220 applying changes to the values in said control-variables spanning each control-variables permissible range and a best polynomial predictor is selected. As discussed in further detail with respect to the Polynomial Optimization And Evaluation Module and FIGURE 7, another simulation process 221 is conducted using control-variable values provided by the selected best polynomial predictor to evaluate the modified values. If an improvement in reactor performance is indicated by the simulation results, the modified control-variables are accepted as an improvement over the initial center-point case. This new combination of independent variables is then re-defined as the new center-point case and the entire control-variable evaluation process is again repeated (as indicated by the dotted line in FIGURE 2) until no further significant improvements are realized. As such the response surface is modified and grown through this process. Once it is determined that no further improvements are obtainable, the response surface is refined using a smaller (more limited) range for control-variable values and the above steps are repeated. The optimization process as a whole is considered essentially completed when no further improvements to the control-variables are discernible and no feasible reduction to the range of control-variable values can be made.

[0031] In FIGURE 3, an overview of an example software system 300 for implementing the multiple control-variable optimization method is illustrated in terms of functionally related sections or "modules" with references to separate accompanying FIGURES 4-8 that show example functional program control steps for each module in greater detail. One or more modules of software system 300, including the software system in its entirety, may be embodied on a computer-readable medium for ease of distribution and installation on one or more processors or networked computer systems. Although sections of functionally related software are described herein in terms of component software modules that may be individually or collectively executed by separate processors, the software system need not necessarily be limited to a modular component implementation. As indicated in FIGURE 3, an example embodiment of software system 300 includes a Response Surface Initialization Module 301, one or more Control-Variable Modules 302, a Polynomial Coefficient Development Module 303, a Polynomial Usage Module 304 and a Response Surface Save/modify Module 305. A modular arrangement of the functionally related software within software system 300 enhances the overall flexibility and applicability of the software system to different environments by facilitating the use or omission of different Control Variable Modules (FIGS. 5A-5E) as desired or appropriate for a particular application and, moreover, facilitates the adding of new and different or updated Control-variable Modules.

[0032] Response surface initialization module 301 is basically responsible for accepting operator-inputted data describing operating conditions and constraints for a given reactor core (e.g., initial core loading, rod pattern, etc.) and creating a starting point or "center-point" simulation case for normalizing response surface 219. Control-variable modules 302 each contain program control steps for creating simulation case data for specific types of reactor core control-variables such as, for example, fuel bundle loading, control rod position, core flow, sequence change locations, bundle characteristics, etc. For each design input (independent control-variable) type, there may be many operational output (independent variable) cases to consider. Furthermore, for each independent variable case considered by a particular control-variable module there are at least two core simulations run from which response data is obtained: one simulation is performed using the center-point simulation case values with the independent control-variable value increased by a predetermined amount and another simulation is performed using the center-point simulation case values with the independent control-variable value decreased by a predetermined amount. The difference between the increased and decreased simulation input values for a particular control-variable or design input is referred to as the range or "breadth" of the control-variable and, since all simulation case results are stored in the response surface, it is also referred to herein as the "breadth" of the response surface (with respect to that control-variable). Each simulation case result includes the values for all of the operational performance parameters (dependent variables) modeled within the core simulation process. Ultimately, the response surface contains at least three core simulation case results for each independent variable case: the center-point case response and two variation case responses created by the particular control-variable module.

[0033] Control-variable modules 302 are preferably executed sequentially using a single computer/processor 10 in the LAN. Additional control-variable modules (not shown here) crafted toward particular reactor plant-specific considerations may also be used. The control-variable modules 302 may be executed in any order and any single one or several control-variable modules may be used (as indicated by the dotted lines in FIGURE 3) depending on the various critical-to-quality considerations and degree of improvement to reactor performance that may be desired. Simulator input data files containing control-variable values are created by each control-variable module and submitted to other computers/processors in the LAN (or remote network 21) that have resident core simulator programs. Once a simulation case is completed by a processor, it creates a simulator output data file containing the resultant values and sends the file to the computer maintaining the response surface. Since reactor core simulations are typically very time consuming, this distributed processing arrangement allows many different core simulation cases to proceed more or less contemporaneously, thereby greatly reducing the overall elapsed time expended on core simulations.

[0034] Alternatively, different control-variable modules could also be resident on different independent computers connected within a LAN, WAN or via other communications links. For example, in such an embodiment, response surface initialization module 301 residing on one computer would place a request over the LAN for the execution of a particular desired control-variable module to another computer on which that module resides and then would forward the center-point case data from the response surface.

[0035] Polynomial coefficient development module 303 contains program control code for mapping the core simulation results for each independent variable case to unique second-order polynomial curves corresponding to each performance parameter (i.e., the operational "dependent" variables). The coefficient values of each polynomial are determined such that each polynomial fits the data from the three simulation cases for its corresponding performance parameter. Polynomial usage module 304 contains program control code for exploring changes to values of each control-variable, as well as changes to combinations of control-variables considered together, and determining which changes produce the greatest impact on core performance. Since running a core simulation is time consuming, the polynomials are used as fast predictors (relative to the 3-D simulator execution) to determine performance parameter values over the input breadth of a control-variable in lieu of running a core simulation. The control-variable(s) having the greatest performance impact are determined by reiteratively comparing predicted performance parameter values using a predetermined objective function. Finally, a Save/modify module 305 contains program control code for saving and documenting the response surface and outputting quantified optimum control-variable operational values or, alternatively, modifying the response surface if it is determined that results can be further improved by reducing the "breadth" of the response surface (explained in greater detail below).

[0036] Referring now to FIGURE 4, a flow chart illustrates example functional steps performed by response surface initialization module 301. The first few initial steps 401-404 basically acquire and identify information needed to create an initial center-point simulation case. At step 401, cycle specific reactor core operating condition data including initial values for control variables (i.e., initial control rod pattern, initial core loading arrangement, etc.) and an initial response surface breadth is specified via operator-input. At step 402, specific operational constraints, which form the design basis, of a particular reactor plant are identified from the acquired operator-input information-- such design basis and constraint information aids in the evaluation of an "objective function", discussed below, that is used to compare the relative quality of alternative solutions. In addition, the computer operator may select an input option, discussed in greater detail below with respect to the Polynomial Optimization And Evaluation Module and FIGURE 7, that permits the effects on reactor performance of a change in the operational value of two or more control-variables to be considered in combination.

[0037] At step 403, the particular independent control-variables (core loading, rod pattern, core flow, sequence exchange, bundle characteristics, etc.) that are to be considered during the optimization are identified based on the acquired operator-input information. At step 404, the fuel bundles to be used within the core are identified and sorted according to reactivity value. Next, at step 405, a core simulation input data file for producing a center-point simulation case is generated and submitted to a resident (or remote) core simulation program. Once the simulation is finished, the results of the simulation are returned in a simulation output file. At step 406, a multidimensional array is created in memory as a simulation "response surface" and data from the simulation output file is stored there as an initial center-point case.

[0038] Next, one or more control-variable modules 302 are executed to develop simulation case data for variations in values for specific control--variables. The execution of more than one control-variable module is optional. As will be readily apparent from this disclosure, additional control-variable specific modules (not disclosed herein) may also be included as desired. As previously mentioned, the individual control-variable modules may be executed sequentially by a single processor or run contemporaneously on different computers within the LAN or WAN. As the execution of each control-variable module results in adding more simulation case data to the response surface, the accuracy of the present method and the potential reactor performance optimization achievable is correspondingly enhanced.

[0039] Referring to FIGURE 5A, the functional steps performed by an example control-variable module for fuel bundle loading are discussed first. The fuel bundle loading module examines changes in reactor performance parameters caused by changes in the fuel bundle position or loading arrangement. Conventionally, most reactor cores are octant-symmetric and, consequently, only bundle arrangements within one octant of the core need to be considered. However, octant symmetry is not a requirement of the process. As indicated at step 501, it is first determined if fuel bundle loading changes are allowed given the pre-identified constraints for the particular reactor. If bundle loading changes are not allowed, program control is passed to another module. If bundle loading changes are allowed, all permissible bundle locations are systematically considered by repeating steps 503 through 507 for each different location, as indicated by block 502.

[0040] At step 503, the known reactivity value of the bundle at the selected location is changed to a predetermined higher value. A new core simulation input file is then generated -- the input file reflecting the change in fuel bundle reactivity value and a shuffling of the remaining fuel to minimize any reactivity differences relative to the center point. This shuffling of the remaining fuel is readily accomplished by referring to the previously sorted list generated by step 404, whereby bundle rank positions in the sorted list are shifted by one position in a 'cascade' strategy. For example, a location that is changed from reactivity rank 10 in the sorted list to rank 5 will have the effect of changing rank 5 to 6, rank 6 to 7, and so forth up until rank 9 to 10. The core simulation input file is then submitted to an available processor/computer for simulation processing, as indicated at step 504. (Although core simulation input files reflecting a "rodded depletion" are generally intended, non-rodded depletion type simulator input files could also be used with this method.) Without waiting for the results of the submitted core simulation, the bundle reactivity value for the same location is changed, at step 505, to a value lower than the original reactivity. The combined amount of increase and decrease exacted to the value for a particular control-variable, as described herein with respect to the various control-variable modules, is predetermined according to the particular control-variable being considered and defines the range or "breadth" of values for which the control-variable is examined.

[0041] Next, at step 506, a new core simulation input file having the changed reactivity value is again generated and submitted to any available processor/computer 10 for processing another simulation. In one operational example, once the simulation cases in steps 504 and 506 are completed, output data parameters from each simulation can be normalized to the center point, fit to polynomials and stored to common response surface 219, for example, by each processor/computer performing the core simulation. If changes in reactivity values for fuel bundles at other locations have not yet been simulated, without necessarily waiting for the core simulations of previous steps to complete, a new bundle location is selected and steps 503-506 are again repeated until all allowable bundle locations have been considered, as indicated at step 507. Ultimately, once all the independent control-variable cases for fuel bundle reactivity variations have been considered, processing may continue under control of another module.

[0042] FIGURE 5B shows program control steps performed by an example control-variable module for the exploring the different axial positions of the control rods or blades. In a manner similar to the fuel bundle loading module of FIGURE 5A, two simulation cases for each control rod are developed and the simulation results are added to the common response surface. At step 509, it is first determined if control rod pattern changes are allowed given the pre-identified constraints for the reactor. If control rod pattern changes are not allowed, program control is passed to another module. If control rod changes are allowed, a predetermined control rod is selected for analysis, as indicated at step 510. Next, at step 511, the initial position value of the selected control rod is increased by a predetermined amount such that the amount of the increase does not violate the physical boundaries of the core or the specified user limits. A new core simulation input file, having only the selected control rod position value changed, is then generated and submitted to an available processor/computer 10 for simulation processing, as indicated at step 512.

[0043] At step 513, the control rod position value for the same control rod is changed to a value less than the original position as was done in step 511. Next at step 514, a new core simulation input file having the changed position value is again generated and submitted to an available processor/computer 10 for processing a second simulation case. As indicated at step 515, if changes in position values for other control rods are to be simulated, a new control rod is selected and steps 511-514 are again repeated until all control rods have been considered. As with the fuel bundle loading module, each step in the control rod positioning module may proceed without necessarily waiting for the core simulations of previous steps to complete. Finally, once all the independent control-variable cases for control rod position variations have been considered, processing may continue under control of another module.

[0044] FIGURE 5C shows program control steps performed by an example control-variable module for developing the response surface from changes in the core flow. In a manner similar to the other independent control-variable modules of FIGURES 5A and 5B, two simulation cases for each core flow control-variable are developed and added to the common response surface. At step 519, it is first determined if core flow changes are allowed given the pre-identified constraints for the reactor. If core flow changes are not allowed, program control is passed to another module. If core flow changes are allowed, a particular core flow variable is selected for analysis, as indicated at step 520. Next, at step 521, the initial center-point case value of the selected core flow variable is increased by a predetermined amount. A new core simulation input file, having only the selected core flow variable value changed, is then generated and submitted to an available processor/computer 10 for simulation processing, as indicated at step 522.

[0045] At step 523, the core flow value for the same core flow variable is changed to a value less than the original value similar to step 521. Next at step 524, a new core simulation input file having the changes core flow value is again generated and submitted to an available processor/computer for processing a second simulation case. As indicated at step 525, if changes in core flow values for other core flow variables have not yet been simulated, the next independent core flow variable is selected and steps 521-524 are again repeated until all independent core flow variables have been considered. As with the other control-variable modules discussed above, each step in this module may proceed without necessarily waiting for the core simulations of previous steps to complete. Finally, once all the independent control-variable cases for core flow variables have been considered, processing may continue under control of another module.

[0046] FIGURE 5D shows program control steps performed by an example control-variable module for developing the response surface from changes in sequence interval. In a manner similar to the other control-variable modules, two simulation cases for each control blade sequence interval occurring during the operational cycle are developed and added to the common response surface 219. At step 529, it is first determined if sequence interval changes are allowed given the pre-identified constraints for the reactor. If changes are not allowed, program control is passed to another module. If changes are allowed, a particular sequence interval is selected for analysis, as indicated at step 530. Next, at step 531, the initial center-point case sequence interval value for the selected sequence interval is increased by a user specified amount. A new core simulation input file, having only the selected sequence interval value changed, is then generated and submitted to an available processor/computer 10 for simulation processing, as indicated at step 532.

[0047] At step 533, the sequence interval value for the same control blade sequence interval is changed to a value less than the original value similar to 531. Next at step 534, a new core simulation input file having the changed position value is again generated and submitted to an available processor/computer for processing a second simulation case. As indicated at step 535, if changes in values for other sequence interval variables have not yet been simulated, a new bundle is selected and steps 531-534 are again repeated until all other relevant independent sequence interval variables have been considered. As with the other control-variable modules, each step in this module may proceed without necessarily waiting for the core simulations of previous steps to complete. Finally, once all the independent control-variable cases for the sequence interval variables have been considered, processing may continue under control of another module.

[0048] Although the modules depicted in FIGURES 5A through 5D together demonstrate the ability of the optimization method to consider independent control-variables that are capable of having values that are considered as "continuous" in nature, such as, for example, loading parameters, rod pattern parameters, flow parameters, and sequence exchange parameters, etc., the method can also be used to consider changes in "discrete" value control-variables, such as bundle characteristics. An example control-variable (CV) module for considering discrete-value type control-variables is provided using the context of fuel bundle characteristics is illustrated in FIGURE 5E.

[0049] Referring now to FIGURE 5E, example program control steps for developing reactor simulation response data from changes in bundle characteristics are described. Fuel bundle characteristics, in this example, can represent any fuel bundle having differences in fuel rod configurations such as that due to radial and/or axial Uranium 235 enrichment variation and/or radial and/or axial Gadolinium variation. Like the previously discussed modules, core simulator cases are generated and executed for each independent control variable. Upon completion of each fuel bundle characteristics independent control-variable, the dependent variable output information is normalized to the relative centerpoint. However, instead of mapping the response to polynomials, the response is mapped to linear functions. Once all control variable modules 302 and corresponding simulation cases have finished execution and the simulation results normalized to the relative center-point, then the simulation case data is mapped to either polynomials and/or linear functions and the results are stored in the response surface 219.

[0050] FIGURE 6 shows example functional program control steps for developing polynomial coefficients for mapping each simulation case to a polynomial that fits the three data values for each independent variable case (i.e., the upper, lower and center-point values). At functional step 601, further processing is delayed until all of the simulation cases are complete and the response surface has been updated. Next, at steps 602 and 603, the response surface is accessed and all the simulation data produced by control variable modules 302 is normalized to the center-point case data. Next, at functional step 604, coefficients are determined for defining a unique second order polynomial that fits the three normalized simulation case values for each independent control-variable. However, since the evaluation of certain control-variables (for example, fuel bundle rod configuration) can only be evaluated as discrete changes, core simulation results for these type of variables are stored in the response surface as discrete first order evaluations and are not mapped to polynomials. Finally, at step 605, the coefficients for each polynomial are saved and further processing continues with the polynomial optimization and evaluation module.

[0051] FIGURE 7 shows example functional program control steps for polynomial optimization and evaluation module 304. This module examines reactor performance parameter values predicted by each of the second-order polynomials associated with each control-variable to determine which control variable and value produces the most significant improvement in reactor performance. At steps 700 and 701, polynomials developed from each of the control-variable simulation cases are accessed from the response surface, sub-grouped and used to predict quantitative values for performance parameters (e.g., CPR, MFLPD, MAPLHGR, etc.) over the breadth of allowable values for that control-variable. In other words, a control-variable is selected and the polynomials associated with each of the performance parameters (i.e., operational outputs) as influenced by that control-variable are used to predict a set of performance parameter values indicative of reactor performance for each of a predetermined number of discrete incremental changes in the value of the selected control-variable over the breadth (i.e., range of predetermined permissible values) of the control-variable. This process is repeated for every independent control-variable.

[0052] Under a principle generally known in the art as "superposition", the net effect of a plurality of changes made to different control-variables together in combination can be determined by the summation of the effects of the individual control-variable changes made separately. Accordingly, at the initialization and input stage (i.e., when cycle specific inputs and design basis considerations are identified, e.g., as discussed above with respect to steps 401 and 402 of the Initialization Module in FIGURE 4), a user of the present system may select an optimization "resolution" level as input option that permits changes to quantitative operational values for more than one independent variable to be evaluated in combination with each other. Consequently, if this option was previously selected, then, at step 700, the individual polynomial-predicted effects of every combination of a selected plurality of independent control-variables are summarily combined to quantitatively determine the net effect that a plurality of changes to different control-variables made together would have on each of the many reactor core performance parameters. The higher the selected resolution level, the more independent control-variables are evaluated together in combination and, hence, the greater the probability of detecting a combination that will improve reactor performance. For example, at a selected optimization resolution level of "three", a change in the quantitative values for three different independent control-variables and every combination of three control-variables out of the total number of control-variables considered would be evaluated. All discrete changes among the plurality of control-variables under a particular resolution are examined using the associated polynomial predictors for each control variable.

[0053] Although higher resolution levels may require somewhat longer processing times than lower resolution levels, the total processing time is significantly less than conventional methods because the polynomial predictors are used and combined accordingly instead of performing actual computer simulations of the reactor core for each case. In this manner, the method is essentially exhaustive and is almost guaranteed to identify the global optimum fuel-cycle design. While very high resolution levels may not be feasible in practice due to the extended processing time required, the capacity of this method to permit selection of a particular resolution level enables the system user to selectively quantify a degree of "closeness" to the true absolute optimum which is desired to be achieved.

[0054] Next, at step 702, for each quantitative value change made to a individual control-variable or combination of control-variables (i.e., the design inputs), an "objective function" test is used to quantify the relative "worth" or "strength" of that change in terms of its effect on improving the performance parameters (i.e., the "dependent" variables). The objective function sets a particular limiting value for each performance parameter that is determined primarily through an integration of performance "violations" relative to defined design limits, offset by the integration of any performance "credits" associated with beneficial results such as additional energy, increased thermal margin, etc. Pre-determined multipliers (i.e., mathematical factors) are applied to design limit values for each of the performance parameters-such as, for example, Hot Excess, MFLPD, MAPLHGR, etc. -- to provide normalization and relative ranking of each parameter. Basically, in step 702, each predicted performance parameter value is tested using an objective function, fashioned in accordance with conventional knowledge and practice in the art, to determine the best set of control-variable polynomial predictors for optimizing core performance. At step 703, the best values for the control-variables are identified. Since each polynomial predictor corresponds to a specific control-variable, polynomial predictors are compared, as rated by the objective function of step 702, and reiteration of steps 700-702 continues until the best values for the control-variables have been identified. Next, at step 704, the control-variable values are compared with the values obtained from previous iterations (if any) to determine if any improvement is found to exist (i.e., improvement in the figure of merit provided by the objective function). If no improvement is detected, processing continues with the steps shown in FIGURE 8. If some improvement is found to exist, a core simulator input case is prepared using the improved values from the selected best polynomial predictor(s) corresponding to one or more control-variables and a core simulation is executed, as indicated at step 705.

[0055] Although the use of polynomials allows for a rapid prediction of what changes may constitute an improvement in reactor performance, the core simulation at step 705 provides calibration between the simulation process and the polynomial coefficient data in the response surface. Essentially, it allows for verifying the predicted improvement by providing "actual" (as opposed to "predicted") core simulation data documenting the operation of the core under the improved control-variables. At step 706, the core simulation results of step 705 are compared with the core simulation results from the center-point case (or the results of previous optimizations) to determine if any improvement to core performance has resulted. If the results from the step 705 core simulation show an improvement over the center-point case, the improvement is incorporated and the process is repeated again, as indicated at step 708. If the results of the core simulation at step 705 have not improved, the corresponding control-variable(s) is considered as "unreliable" and marked as such at step 707. Namely, in step 707 the corresponding control-variable values will not be considered as a potential solution. Once a predetermined number of unreliable control-variables is exceeded, as tested at step 709, polynomial optimization and evaluation ceases and processing continues with the steps shown in FIGURE 8.

[0056] FIGURE 8 shows example functional program control steps for Save/modify Response Surface Module 305. First, the current "breadth" of the response surface is examined at step 801 (i.e., the breadth of the response surface in terms of the range of control-variable values explored). If a reduction in the predetermined range of values used by the CV modules in creating simulation cases for the control-variables is feasible, then that range is decreased and the creation of a new response surface is initiated using the original center-point case data. This is indicated at functional step 802 as reducing the response surface breadth. At this point, the optimization process starts over again creating this "new" response surface using one or more of the various control-variable modules, as indicated by entry point "B" in FIGURE 4. If reducing the "breadth" of control-variable values used by the CV modules is not feasible, the current response surface data is documented (saved) and the optimized control-variable values are output, as indicated by steps 803 and 804.

In-Cycle Shuffling

[0057] FIGURE 9 illustrates a flow chart of the method of improving reactor performance through in-cycle shuffling according to an embodiment of the present invention. For the purposes of explanation only, the method of improving reactor performance through in-cycle shuffling according to FIGURE 9 will be described as being implemented by the architecture illustrated in FIGURE 1B. As shown, in step S30 an initial operating plan is developed as discussed in detail above. However, this initial operating plan is developed knowing a priori that and in-cycle shut down and shuffling operation will take place. In this situation, the initial operating strategies developed in step S30 may differ from typical historical concepts. For example, an operating plan designer may target less operating margin for the part of the cycle following the in-cycle shuffle. Because the loading and rod pattern strategy may be modified at the time of the in-cycle shuffle, additional operating margin may be included during the in-cycle shuffle development. A user then models operation of the core to a user selected in-cycle shut down point such as mid-cycle in step S32. Namely, the user runs a well-known simulation program on the developed core design to simulate operation up to the in-cycle shut-down point. Depending on the results of the simulation, the user may decide to re-run the simulation for a different in-cycle shut-down point. Factors that would influence the time of the in-cycle shuffle would be the expected price of energy at the time of the in-cycle outage, the availability of the outage crews (some utilities own multiple reactors that utilize the same outage crew), etc. Step S32 may be performed at any point in time prior to the user selected in-cycle shut-down point provided sufficient time is available to license and implement the new operating plan based on shuffling the core as explained in detail below. For example, step S32 may be performed by the user prior to actual implementation of the core design in the reactor or sometime after the reactor has been operating according to the core design.

[0058] Assuming, that step S32 is performed after the reactor has been operating according to the core design, then many of the actual independent control variable values and resulting dependent variable values may have deviated from the core design. Accordingly, these variations may be incorporated into the simulation run in step S32. The independent control variable values and dependent variable values determined at the user selected in-cycle shut-down point represent the predicted state of the reactor at the in-cycle shut-down point.

[0059] An optimized operation plan for reactor core operation from the in-cycle shut-down point until the end of cycle is then developed in step S34. This operation is performed in the same manner that the initial operation plan was developed in Step S30, except that the predicted state of the reactor at the in-cycle shut-down point is used as the initial independent control variable values and dependent variable values for the operating plan optimization process. Furthermore, the operating plan optimization process described in detail above will be performed for many possible permutations of fuel bundle shuffling. As will be appreciated, fuel bundle shuffling is the moving of bundles in the core to different positions in the core. Accordingly, the possible permutations include not only the number of fuel bundles being moved, but also the different positions that those bundles may be moved to. Alternatively, the number of possible fuel bundle shuffling case scenarios may be constrained based on user entered constraints such as a minimum number of fuel bundles shuffled, a maximum number of fuel bundles shuffled, a maximum distance that a fuel bundle may be moved, etc. In this alternative embodiment, an optimal operating plan is developed only for those fuel bundle shuffling case scenarios meeting the user constraints. In a further alternative, the fuel bundle shuffling case scenarios are specified by the user.

[0060] Once the candidate solutions have been generated for each fuel bundle shuffling case scenario, the best solution is determined in step S36. The best solution is the solution that produces the greatest amount of energy while meeting operating margins, and thus, will result in the greatest amount of revenue.

[0061] In step S40, the user determines whether this best solution is acceptable; namely, the user determines if the best solution generates more energy than continuing reactor operation without a shut down and core shuffle. As will be appreciated, in making this decision, the user will account for the cost of shutting down the reactor. If the best solution is unacceptable, then the best solution is not implemented and reactor operation continues without shut down and core shuffling.

[0062] However, having determined an acceptable solution, the resulting core design is licensed in the well-known manner in step S43. Subsequently, the reactor is shut down in step S44, and the newly licensed core design is implemented in step S46. Reactor operation will then resume according to the new core design in step S48.

[0063] As discussed previously, the inventors have discovered that an in-cycle shutdown and shuffling operation as described above can increase energy output over the cycle, and thus increase the revenue generated by the reactor.

[0064] The inventors have further determined that additional benefits may be derived by the in-cycle shuffling process. The planned in-cycle shuffling process could be used to mitigate any unanticipated issues related to channel bow, control blade history and most commonly failed fuel. A severe channel bow problem could limit control blade movement or in a worst scenario completely disable the control blade. The failed fuel is a fuel bundle assembly that may begin to leak during the normal operation of a reactor. This occurs when a hole (small or large) has been made in one or more of the individual fuel assembly rods. Therefore, the interior of the fuel assembly rod, which contains a variety of radioactive elements, is exposed to the reactor vessel water. This release of radioactive materials to the larger core vessel can be environmentally, and economically disastrous. There are numerous potential causes for such holes or cracks. They can be caused by poor manufacturing welds of the fuel pin, poor blends of the cladding material, debris in the water that constantly rubs against the cladding to create an opening, corrosion caused by poor water chemistry, etc. Operational occurrences such as rapid rod pattern adjustments have also been known to cause damage to the fuel assembly rods. Control blade history is a well-know nuclear reaction phenomenon that can adversely impact the fuel bundle thermal performance for those fuel bundles located in the vicinity of the control blade. The ultimate goal is to limit the controlled exposure to below a certain critical value thereby assuring no degradation in fuel bundle thermal performance.

[0065] Whenever, the rod clad integrity is breached and the radioactive materials are exposed to the reactor water, trace elements of radioactive materials are present in the water and can be detected by water samples. These samples are often referred to as "offgas." Once the offgas identifies higher than usual amounts of the radioactive components, it is known that there are one or more fuel bundle assemblies that contain leaking fuel rods. After such occurrences, a systematic process of suppressing and measuring offgas amounts is done at reduced power levels in order to try to identify the location and number of leaking fuel bundle assemblies. In addition, slight differences of radioactive combinations can provide evidence as to the age of the leaking fuel bundle assembly. Hence, without opening the core plant, managers and engineers can obtain fair estimates of the suspect leaking bundles and their locations. With this information, possible solutions can be studied.

[0066] Historically, there have been four main solution potentials. First, the cycle could prematurely end. This would typically be the case if the offgas levels were very high and/or the cycle was close to EOC already. Secondly, the plant could decide to do nothing. As a result of this action, offgas levels would most likely remain steady or exponentially increase as operation continues. This is typically only a temporary measure and is used while offgas levels are acceptably low. A third potential is to suppress the areas of the core around the leaking fuel assemblies with control blades. In doing so, the amount of power produced by the leaking fuel assemblies is lowered. This lowering of power can minimize the duty of assemblies. Although this will not solve the problem, it can increase the time the reactor can maintain adequate power levels and minimize damage to the leaking fuel bundle assemblies. Consequently, offgas levels will increase at a much smaller rate. Such an approach often buys a couple of months of additional operation but at greatly reduced full cycle capability and margin for thermal limit considerations. This path is typically an interim step taken until a planned outage can be arranged. A fourth solution is to shutdown the reactor core, and replace the leaking fuel bundle assembly. Once the damaged fuel bundle assemblies are replaced with bundles of similar or lesser reactivity, the reactor core is restarted. Although this solution solves the leaking fuel assembly problem, this solution is very costly to the reactor operating company. While the reactor was shutdown, no power was produced; and therefore, no revenue earned. Furthermore, there is a large cost associated with the shutdown, replacement and restart process. Often, several hundred to over a thousand people are involved during a reactor outage. Besides a leaking fuel rod, other problem situations such as a variety of equipment failures (turbine, pump, instrumentation, etc) can arise that also require a core shutdown, with the same associated loss of revenue and high costs.

[0067] The inventors have discovered that when these problems arise in the context of a reactor core having an operating plan that includes a scheduled in-cycle shutdown, these problems may be eliminated during the core shuffling process and proactively improve the cycle performance. In addition, the optimization of the operating plan may account for the elimination of problem bundles and the inclusion of replacement bundles.

[0068] FIGURE 10 illustrates an embodiment of this invention. As shown, the embodiment of FIGURE 10 is the same as the embodiment illustrated in FIGURE 9 except for additional step S33. Accordingly, only these additional steps will be described for the sake of brevity. As shown, after modeling operation of the nuclear reactor to the user selected in-cycle shutdown point. In step S33, the user determines which fuel assembly bundles to discharge during the shutdown. For example, when the problem encountered is a leaking fuel assembly bundle or bundles, these bundles will be selected for discharge. In addition, bundles of the similar type in symmetric positions may also be targeted for discharge. The selection of discharged bundles can also include bundles adjacent to the leaking fuel assembly bundles, and bundles where leaks are suspected but not confirmed. The process of determining which fuel assembly bundles to discharge will typically involve input from the personnel operating and managing the reactor. Either based on experience, historical data, or simple budgetary constraints, these personnel will often direct which bundles are replaced, and place limits as to which bundles can be replaced.

[0069] Subsequently, the user specifies the replacement set of fuel bundles. The replacement set may include fresh fuel bundles selected by the user or fuel bundles residing in the fuel pool of the nuclear reactor or another nuclear reactor's fuel pool. Here, the user specifies each fuel bundle replacing a fuel bundle being removed. The method then continues on to step S34.

[0070] The inventors further discovered that even if no problem bundles exist, the in-cycle shut-down and core shuffling process may result in further benefits if the user includes the fuel pool of the nuclear reactor or another reactor's fuel pool in the shuffling process. Namely, the user may move fuel bundles into the core from the fuel pool and out of the core into the fuel pool.

[0071] The invention being thus described, it will be obvious that the same may be varied in many ways. Such variations are not to be regarded as a departure from the spirit and scope of the invention, and all such modifications as would be obvious to one skilled in the art are intended to be included within the scope of the following claims.